Flask User Interface Report

User Interface Design (Flask)

The user interface for this projectis developed using Flask, a lightweight Python web framework.

The goal is to provide a simple and clear way for users to enter reviews and compare predictions from two models trained on balanced and imbalanced datasets. Key Design Elements:

1. Input Form: A single text area is provided in index.html where the user can type a review. This allows flexibility for handling real-world review lengths.

2. Predict Button: Submitting the form sends the review text to the Flask backend (app.py) through a POST request. This ensures predictions are only generated when explicitly requested.

3. Comparison Output: The predictions from both models are displayed in separate boxes within the same results page. This side-by-side design helps the user compare the outputs easily.

4. Error Handling: If the input is empty, the system displays a warning instead of attempting prediction. Flask also checks whether the models and vectorizers are correctly loaded before prediction.

5. Project Structure: - app.py → Main Flask backend. - templates/index.html → Frontend interface (form + results). - app/ → Contains model pipelines (model_A_pipeline.pkl, model_B_pipeline.pkl).

User Flow (Flask)

1.The userlaunchesthe Flask app locally (python app/app.py) and opens it in a browser (http://127.0.0.1:5000/).

2. The user enters a product review in the input form.

3. On clicking Predict, Flask receives the input and passes it to both pipelines.

4. Each pipeline (balanced and imbalanced) processes the text and generates a prediction.

5. Flask renders index.html with the results: - Prediction from Model A (Balanced). - Prediction from Model B (Imbalanced).

6. The user can repeat the process with any number of reviews.